UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/538,152	06/08/2005	Marie-Catherine Fritsch	2002P18325WOUS	3257

7590 12/24/2009

Siemens Corporatio
Intellectual Property Department
170 Wood Avenue South
Iselin, NJ 08830

EXAMINER	
SINGH, HIRDEPAL	

ART UNIT	PAPER NUMBER
2611	

MAIL DATE	DELIVERY MODE
12/24/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/538,152
Filing Date: June 08, 2005
Appellant(s): FRITSCH ET AL.

Janet D. Hood
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 9/24/2009 appealing from the Office action mailed 5/29/2009.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,805,896	**Burgess**	**9-1998**
6,334,076	**Sakurai**	**12-2001**
2002/0165744	**Juras**	**11-2002**
6,023,702	**Leisten**	**2-2000**

Elmqvist "A Uniform Architecture for Distributed Automation" Advances in Instrumentation and Control, Research Triangle Park, NC US, Vol. 46, Part 2 (1991), Pages1599-1608

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

The rejections are hereby reproduced for convenience.

Claim Objections

1. Claim 35 is objected to because of the following informalities: claim 35 recite limitations including some S and P parameters, without defining what type of parameters they are and what values they can acquire. Appropriate correction is required.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 13, 26 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burgess (US 5,805,896), in view of Sakurai et al. (US 6,334,076), in view of Juras et al. (US 2002/0165744) and further in view of Elmqvist ("A Uniform Architecture for distributed automation", Advances in Instrumentation and Control, Instrument Society of America, Research Triangle Park, NC US, Vol. 46, Part 2, 1991; Pages, 1599-1608).

Regarding Claims 13 and 26:

Burgess discloses a system and method for producing software/code using links of the components of the system (summary of the invention) comprising:

sending messages between the components through the ports and the data is being transferred between the components (column 2, lines 23-30), therefore it is inherent that the message transfer is taking place as signals through the ports;

the event objects include message information describing the message i.e. information about information, and the derived class provides behavior specific to a type of message i.e. message is the information and type of message is metainformation i.e. information about information (column 2, lines 23-40), also the system components are sending and receiving the temperature data and also converting from one scale to

another i.e. Fahrenheit to Centigrade and vice versa (figures 4-7; column 3, lines 20-58), in this case the temperature data is the information and the information whether the temperature scale in Fahrenheit or Centigrade is metainformation i.e. information about information;

producing a program code by interconnecting the signals based on the directed connections of the components (column 4, lines 35-50; producing a class is referred to as a program code; system shields (protects) the programmer or developer from details of connecting of components see column 2, lines 18-21; also in the system the message is sent from dispatching to target member, see column 2, lines 28-32)

Burgess discloses all of the subject matter as described above and further discloses that the components have input and output ports, represented by corresponding symbols/functional blocks/modules (column 1, lines 45-64; column 2, lines65-67; column 3, lines 1-19) and; the components are connected through their ports, directed relationship of the components are defined (column 3, lines 29-34, lines 54-57; column 4, lines 1-16), but doesn't specifically teach that (1) the code generation is for a manufacturing and/or processing plants, and the automation code is generated on the basis of a structure of the plant and know how, including predecessor/successor relationship (similar to directed relationship in Burgess), previously input into the description; (2) the components are described in drawing comprising control relevant information in the manufacturing and/or processing plant; (3) the control information described in the drawing is based on the material flow in the manufacturing and/or processing plant.

Regarding item (1) above, Examiner notes that code generation for a manufacturing and/or processing plants this is just an intended use, therefore little if any patentable weight is given.

However, regarding item (1), Sakurai in the same field of endeavor discloses a similar system and method for automatically generating a control program/code for plants such as rolling plants, power plants, and chemical plants (abstract, technical field); and the automation code is generated on the basis of a structure of the plant and know how previously input into the description (plant operation procedure is defined in a flow chart (SFC), and sequential control flow is defined in a chart and block diagram see column 4, lines 36-48; which means the information about the operation and process in the plant in defined in a chart and used for generation of automation code; also figure 10; column 10, lines 60-67); regarding item (2) above, Sakurai discloses that the components of the system are represented by functional modules in form of drawings or pictures or graphics based on the control relevant information i.e. operation procedure, and the system is controlled by modifying the drawings or graphics or pictures of the described component modules (column 2, lines 20-51) where a picture representative of the plant operation control specification entered for the generation of a program can be viewed on crt (column 4, lines 8-22).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to use the disclosed system for code generation by Sakurai in Burgess in a manufacturing and/or processing plant to generate automation code for controlling a manufacturing and/or process plant based on the plant layout and a relationship of

component how the process flows in a plant to generate the automation code to allow a person with little programming knowledge to generate the code, and to make system capable of checking and modifying the function of automatically generated code before the plant is caused to run actually based on the information available about the plant or factory to reduce errors that may be caused by choosing a different or reverse direction of process or material flow in the plant.

Regarding item (3) above, Elmqvist discloses a similar system and method for distributed automation with a graphical programming environment for programming/software generation by graphically connecting the predefined modules (abstract, page 1599; paragraph 4, page 1600), and further discloses that the control information in drawing or graphic is based on the physical objects present in the processing or manufacturing plant as pumps, pump stations, robots, roller tables etc. (paragraph; Object and data flow based language, page 1600). This is inherent that the physical objects of the plant form the path for material or fluid flow as shown in the example of tank system (figures 1-5) i.e. the system is controlling the process based on the material or fluid flow through the tanks, PID (process identifier) controllers, valves, and pumps (Tank system, page 1601).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to use a drawing or picture or graphic having control relevant information based on material flow in a plant for code generation in Burgess in order to combine the graphically represented components i.e. a drawing based on material flow

in a plant of Elmqvist for code generation to help make use of the standard designing tools.

Regarding the Predecessor/successor relationships to make rejection clear following reference is used, Juras in the same field of endeavor discloses a system and method for product development process where the software code is generated by using plant layout for building manufacturing systems (figures 12-16; paragraphs 0036 and 0051)

Therefore, it would have been obvious to one of ordinary skill in the art the teachings of Juras in the Burgess system for predecessor/successor relationships for a finite number of identified, predictable solutions with reasonable success i.e. to implement the predecessor/successor relationships in the manufacturing plant to define the relation between components of the system as predecessor/successor relationships in burgess system in order to get the proper order for the execution of the program based on the priority of the process.

Regarding Claim 33:

Burgess discloses a system and method for producing software/code using links of the components of the system (summary of the invention) comprising:

the components of the system have input and output ports for data or message communication (column 1, lines 45-64; column 2, lines65-67; column 3, lines 1-19);

the components are connected through their ports for communicating or sending/receiving messages i.e. a communication network between the components of the system, and a controller i.e. a class object controls the communication of messages

between the components (column 4, lines 1-50) and the components are connected through their ports, directed relationship of the components are defined (column 3, lines 29-34, lines 54-57; column 4, lines 1-16);

the components have input and output ports, represented by corresponding symbols/functional blocks/modules (column 1, lines 45-64; column 2, lines65-67; column 3, lines 1-19), and the components are connected through their ports, direction of the connection is indicated between input and output ports (column 3, lines 29-34, lines 54-57; column 4, lines 1-16);

producing a program code for the processing or manufacturing plant based on the control information flow and the directed connections of the components (column 4, lines 35-50; producing a class is referred to as a program code).

Burgess discloses all of the subject matter as described above except for specifically teaching that (1) the code generation is for a manufacturing and/or processing plants; (2) the described components of the plant comprising function module and the function module being a reusable software object that defines characteristics and functions of the elements of the plant; and (3) the components are described in drawing comprising control relevant information based on material flow in the manufacturing and/or processing plant.

Regarding item (1) above, Examiner notes that this is just an intended use, therefore little if any patentable weight is given.

Sakurai in the same field of endeavor discloses a similar system and method for automatically generating a control program/code for plants such as rolling plants, power

plants, and chemical plants (abstract, technical field); and the automation code is generated on the basis of a structure of the plant and know how previously input into the description (figure 10; column 10, lines 60-67);

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the disclosed system for code generation in Burgess in a manufacturing and/or processing plant to generate automation code for controlling a manufacturing and/or process plant to allow a person with no programming knowledge to generate the code, and to make system capable of checking and modifying the function of automatically generated code.

Regarding item (2) above, Sakurai discloses a similar system and method for automatically generating a control program/code for plants such as rolling plants, power plants, and chemical plants as above, and further discloses that the components of the system are represented by functional modules, and the function modules are reusable or the combination of modules is selected according to the operation and procedure of the plant (column 2, lines 20-51).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the function module of the components of plant with connections for communication, as reusable software object for code generation in Burgess to combine the function module as reusable software code, defining functions and characteristics of elements of the plant for code generation to help make use of the standard designing tools.

Regarding item (3) above, Sakurai discloses that the components of the system are represented by functional modules in form of drawings or pictures or graphics based on the control relevant information i.e. operation procedure, and the system is controlled by modifying the drawings or graphics or pictures of the described component modules (column 2, lines 20-51). Furthermore, Elmqvist discloses a similar system and method for distributed automation with a graphical programming environment for software generation by graphically connecting the predefined modules (abstract, page 1599; paragraph 4, page 1600), and further discloses that the control information in drawing or graphic is based on the physical objects present in the processing or manufacturing plant as pumps, pump stations, robots, roller tables etc. (paragraph; Object and data flow based language, page 1600). This is inherent that the physical objects of the plant form the path for material or fluid flow as shown in the example of tank system (figures 1-5) i.e. the system is controlling the process based on the material or fluid flow through the tanks, PID (process identifier) controllers, valves, and pumps (Tank system, page 1601).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use a drawing or picture or graphic having control relevant information based on material flow in a plant for code generation in Burgess to combine the graphically represented components i.e. a drawing based on material flow in a plant of Elmqvist for code generation to help make use of the standard designing tools.

Regarding the Predecessor/successor relationships to make rejection clear following reference is used, Juras in the same field of endeavor discloses a system and method for product development process where the software code is generated by using plant layout for building manufacturing systems (figures 12-16; paragraphs 0036 and 0051)

Therefore, it would have been obvious to one of ordinary skill in the art the teachings of Juras in the Burgess system for predecessor/successor relationships for a finite number of identified, predictable solutions with reasonable success i.e. to implement the predecessor/successor relationships in the manufacturing plant to define the relation between components of the system as predecessor/successor relationships in burgess system in order to get the proper order for the execution of the program based on the priority of the process.

4. Claims 13, 17, 19, 23, 26, 29, 31, 33 and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burgess (US 5,805,896), in view of Sakurai et al. (US 6,334,076), further in view of Elmqvist ("A Uniform Architecture for distributed automation", Advances in Instrumentation and Control, Instrument Society of America, Research Triangle Park, NC US, Vol. 46, Part 2, 1991; Pages, 1599-1608) and in view of Leisten et al. (US 6,023,702).

Regarding Claims 13 and 26:

Burgess discloses a system and method for producing software/code using links of the components of the system (summary of the invention) comprising:

sending messages between the components through the ports and the data is being transferred between the components (column 2, lines 23-30), therefore it is inherent that the message transfer is taking place as signals through the ports;

the event objects include message information describing the message i.e. information about information, and the derived class provides behavior specific to a type of message i.e. message is the information and type of message is metainformation i.e. information about information (column 2, lines 23-40), also the system components are sending and receiving the temperature data and also converting from one scale to another i.e. Fahrenheit to Centigrade and vice versa (figures 4-7; column 3, lines 20-58), in this case the temperature data is the information and the information whether the temperature scale in Fahrenheit or Centigrade is metainformation i.e. information about information;

producing a program code by interconnecting the signals based on the directed connections of the components (column 4, lines 35-50; producing a class is referred to as a program code; system shields (protects) the programmer or developer from details of connecting of components see column 2, lines 18-21; also in the system the message is sent from dispatching to target member, see column 2, lines 28-32)

Burgess discloses all of the subject matter as described above and further discloses that the components have input and output ports, represented by corresponding symbols/functional blocks/modules (column 1, lines 45-64; column 2, lines65-67; column 3, lines 1-19) and; the components are connected through their ports, directed relationship of the components are defined (column 3, lines 29-34, lines

54-57; column 4, lines 1-16), but doesn’t specifically teach that (1) the code generation is for a manufacturing and/or processing plants, and the automation code is generated on the basis of a structure of the plant and know how, including predecessor/successor relationship (similar to directed relationship in Burgess), previously input into the description; (2) the components are described in drawing comprising control relevant information in the manufacturing and/or processing plant; (3) the control information described in the drawing is based on the material flow in the manufacturing and/or processing plant.

Regarding item (1) above, Examiner notes that code generation for a manufacturing and/or processing plants this is just an intended use, therefore little if any patentable weight is given.

However, regarding item (1), Sakurai in the same field of endeavor discloses a similar system and method for automatically generating a control program/code for plants such as rolling plants, power plants, and chemical plants (abstract, technical field); and the automation code is generated on the basis of a structure of the plant and know how previously input into the description (plant operation procedure is defined in a flow chart (SFC), and sequential control flow is defined in a chart and block diagram see column 4, lines 36-48; which means the information about the operation and process in the plant in defined in a chart and used for generation of automation code; also figure 10; column 10, lines 60-67); regarding item (2) above, Sakurai discloses that the components of the system are represented by functional modules in form of drawings or pictures or graphics based on the control relevant information i.e. operation procedure,

and the system is controlled by modifying the drawings or graphics or pictures of the described component modules (column 2, lines 20-51) where a picture representative of the plant operation control specification entered for the generation of a program can be viewed on crt (column 4, lines 8-22).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to use the disclosed system for code generation by Sakurai in Burgess in a manufacturing and/or processing plant to generate automation code for controlling a manufacturing and/or process plant based on the plant layout and a relationship of component how the process flows in a plant to generate the automation code to allow a person with little programming knowledge to generate the code, and to make system capable of checking and modifying the function of automatically generated code before the plant is caused to run actually based on the information available about the plant or factory to reduce errors that may be caused by choosing a different or reverse direction of process or material flow in the plant.

Regarding item (3) above, Elmqvist discloses a similar system and method for distributed automation with a graphical programming environment for programming/software generation by graphically connecting the predefined modules (abstract, page 1599; paragraph 4, page 1600), and further discloses that the control information in drawing or graphic is based on the physical objects present in the processing or manufacturing plant as pumps, pump stations, robots, roller tables etc. (paragraph; Object and data flow based language, page 1600). This is inherent that the physical objects of the plant form the path for material or fluid flow as shown in the

example of tank system (figures 1-5) i.e. the system is controlling the process based on the material or fluid flow through the tanks, PID (process identifier) controllers, valves, and pumps (Tank system, page 1601).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to use a drawing or picture or graphic having control relevant information based on material flow in a plant for code generation in Burgess in order to combine the graphically represented components i.e. a drawing based on material flow in a plant of Elmqvist for code generation to help make use of the standard designing tools.

Regarding the Predecessor/successor relationships to make rejection clear following reference is used; Leisten discloses a system and method of computer system for process and project management for design and manufacturing a product in a plant (column1, lines 15-18) where information for the code generation describing activity types or processes or controls is defined and the directed relationships between predecessor and successor activities are always well defined (column 20, lines 40-48)

Therefore, it would have been obvious to try, to one of ordinary skill in the art the teachings of Leisten in the Burgess system for predecessor/successor relationships for a finite number of identified, predictable solutions with reasonable success i.e. to implement the predecessor/successor relationships in the manufacturing plant to define the relation between components of the system as predecessor/successor relationships in burgess system in order to get the proper order for the execution of the program based on the priority of the process.

Regarding Claims 17 and 29:

Burgess discloses all of the subject matter as described above and further discloses an input device/means for inputting relevant information for producing software code (column 14, lines12-18; fig 9).

Regarding Claim 19:

Burgess discloses all of the subject matter as described above except for specifically teaching that the method for distributed automation with graphical connection represents information flow, and a data flow model.

Elmqvist in the same field of endeavor discloses that the method for distributed automation with graphical connection represent information flow, and a data flow model (page 1601, paragraph 4; page 1605, paragraph 10).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the material flow, and/or energy flow, and/or information flow as a basis for mapping the directed relationships between the components in Burgess system in order to use the material flow, and/or energy flow, and/or information flow as a basis for mapping the directed relationships between the components to make the automation code more effective and error free as the manufacturing and/or processing plant layout and planning is according to the material flow, and/or energy flow, and/or information flow.

Regarding Claims 23 and 31:

Burgess discloses all of the subject matter as described above except for specifically teaching that the system and method is for distributed automation with

automated cooperation for distributed objects; and the system could be a central system.

Elmqvist in the same field of endeavor discloses that the system and method is for distributed automation with automated cooperation for distributed objects (page 1599, abstract paragraph 2; page 1605, paragraph 5). However, official notice is taken that it is old and well known within the computer art that if automated code generation is used for distributed system then it could be used for central system too.

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the disclosed system in Burgess for central and/or distributed solutions to use the disclosed system for central and/or distributed solutions to control the distributed components with a central controller or to control the components with a central controller as required.

Regarding Claim 33:

Burgess discloses a system and method for producing software/code using links of the components of the system (summary of the invention) comprising:

the components of the system have input and output ports for data or message communication (column 1, lines 45-64; column 2, lines65-67; column 3, lines 1-19);

the components are connected through their ports for communicating or sending/receiving messages i.e. a communication network between the components of the system, and a controller i.e. a class object controls the communication of messages between the components (column 4, lines 1-50) and the components are connected

through their ports, directed relationship of the components are defined (column 3, lines 29-34, lines 54-57; column 4, lines 1-16);

the components have input and output ports, represented by corresponding symbols/functional blocks/modules (column 1, lines 45-64; column 2, lines65-67; column 3, lines 1-19), and the components are connected through their ports, direction of the connection is indicated between input and output ports (column 3, lines 29-34, lines 54-57; column 4, lines 1-16);

producing a program code for the processing or manufacturing plant based on the control information flow and the directed connections of the components (column 4, lines 35-50; producing a class is referred to as a program code).

Burgess discloses all of the subject matter as described above except for specifically teaching that (1) the code generation is for a manufacturing and/or processing plants; (2) the described components of the plant comprising function module and the function module being a reusable software object that defines characteristics and functions of the elements of the plant; and (3) the components are described in drawing comprising control relevant information based on material flow in the manufacturing and/or processing plant.

Regarding item (1) above, Examiner notes that this is just an intended use, therefore little if any patentable weight is given.

Sakurai in the same field of endeavor discloses a similar system and method for automatically generating a control program/code for plants such as rolling plants, power plants, and chemical plants (abstract, technical field); and the automation code is

generated on the basis of a structure of the plant and know how previously input into the description (figure 10; column 10, lines 60-67);

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the disclosed system for code generation in Burgess in a manufacturing and/or processing plant to generate automation code for controlling a manufacturing and/or process plant to allow a person with no programming knowledge to generate the code, and to make system capable of checking and modifying the function of automatically generated code.

Regarding item (2) above, Sakurai discloses a similar system and method for automatically generating a control program/code for plants such as rolling plants, power plants, and chemical plants as above, and further discloses that the components of the system are represented by functional modules, and the function modules are reusable or the combination of modules is selected according to the operation and procedure of the plant (column 2, lines 20-51).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the function module of the components of plant with connections for communication, as reusable software object for code generation in Burgess to combine the function module as reusable software code, defining functions and characteristics of elements of the plant for code generation to help make use of the standard designing tools.

Regarding item (3) above, Sakurai discloses that the components of the system are represented by functional modules in form of drawings or pictures or graphics based

on the control relevant information i.e. operation procedure, and the system is controlled by modifying the drawings or graphics or pictures of the described component modules (column 2, lines 20-51). Furthermore, Elmqvist discloses a similar system and method for distributed automation with a graphical programming environment for software generation by graphically connecting the predefined modules (abstract, page 1599; paragraph 4, page 1600), and further discloses that the control information in drawing or graphic is based on the physical objects present in the processing or manufacturing plant as pumps, pump stations, robots, roller tables etc. (paragraph; Object and data flow based language, page 1600). This is inherent that the physical objects of the plant form the path for material or fluid flow as shown in the example of tank system (figures 1-5) i.e. the system is controlling the process based on the material or fluid flow through the tanks, PID (process identifier) controllers, valves, and pumps (Tank system, page 1601).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use a drawing or picture or graphic having control relevant information based on material flow in a plant for code generation in Burgess to combine the graphically represented components i.e. a drawing based on material flow in a plant of Elmqvist for code generation to help make use of the standard designing tools.

Regarding the Predecessor/successor relationships, to make rejection clear following reference is used; Leisten discloses a system and method of computer system for process and project management for design and manufacturing a product in a plant

(column1, lines 15-18) where information for the code generation describing activity types or processes or controls is defined and the directed relationships between predecessor and successor activities are always well defined (column 20, lines 40-48)

Therefore, it would have been obvious to try, to one of ordinary skill in the art the teachings of Leisten in the Burgess system for predecessor/successor relationships for a finite number of identified, predictable solutions with reasonable success i.e. to implement the predecessor/successor relationships in the manufacturing plant to define the relation between components of the system as predecessor/successor relationships in burgess system in order to get the proper order for the execution of the program based on the priority of the process.

Regarding Claim 34:

Burgess discloses all of the subject matter as described above except for specifically teaching that the control system comprises different zones with subsets of plant elements.

Elmqvist in the same field of endeavor discloses that the control system comprises different zones with subsets of plant elements i.e. the tank system with tank 1, PID 1 is a control zone with PID, valve as subset of elements of system, and PID controller work as the control coordinator as shown in the topology of the network of the system (figures 1-3; pages 1602-1603).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to implement the software code generation of Burgess in a system with different control zones with plant elements including controllers. One

would have been motivated to implement the generated code in a system with different control zones including plant elements and controllers to make all different components of system work in coordination for optimum results and control.

(10) Response to Argument

A. with respect to 35 USC 112 omnibus claims

The arguments offered by Appellant regarding Omnibus claims are persuasive. Therefore, the rejection is withdrawn.

B. with respect to indefinite claim language

Appellant's argument about indefinite language is considered and therefore, the 35 USC 112 rejection is withdrawn.

C. with respect to 35 USC 103 rejection

I. predecessor/successor relationships: The argument offered by the Appellant that Burgess does not include predecessor/successor relationship and the cited column 2, lines 23-30 have not mentioned ports, is respectfully traversed because in Burgess's system the message is sent from dispatching to target member (column 2, lines 28-32) i.e. the connections are defined which means who sends and who receives the message (equivalently described on page 8 of present invention paragraph 0020,

lines 21-24). Secondly the Appellant is reminded that the rejection is made based on the entire content of the prior art not just the cited part, therefore Burgess describes the directed relation of ports many times including column 3, lines 1-6 and lines 54-57. The argument regarding predecessor/successor relation not defined in Burgess is moot, first it is described in Burgess clearly, as discussed above and secondly a secondary reference is used for this purpose.

II. Previously input plant structure and know how: Applicant argument about the intended use is a misunderstanding of the rejected limitation, in the previous office actions the intended use is about the generation of automation code in a manufacturing plant as pointed out in last office action. Appellant argument that Sakurai provides standard program and a plant operator input plant operation procedure by keyboard, is a kind of agreement to the examiner's position for rejection i.e. Appellant agreed that cited part of Sakurai for plant operation procedure is input by a user in the system is equivalent to present invention (as in appellant words that, an attempt was made to amend the claims that the plant layout is input into system by a user, as argued in part B) which makes it clear that in both cases i.e. in present invention and in Sakurai the information is input into system by a user makes the rejection clear.

Further Appellant states that Sakurai system is more complex as it requires an operator, but at the same time Appellant agreed that a user is required to input drawing in the system (by attempted after final amendment) therefore agreeing for equal complexity of the reference and instant invention.

III. Description in a drawing: Appellant's argument that Sakurai do not mention a drawing at all is respectfully traversed as previously described in Sakurai system the inputted plant operation procedure is defined in a flow chart the plant operation and sequential control flow is defined in a chart and block diagram (column 4, lines 36-48; that is substantially similar to present invention disclosure page 5, paragraph 0014 see specification file June 8, 2005, as stating that information may be in a tabular form or any other form) which means the information about the operation and process in the plant in defined in a chart and used for generation of automation code and Sakurai column 4, lines 36-42 describes plant operation procedure specified in a flow chart and block diagram.

IV. Description based on material flow: The argument that in Elmqvist system the process order could be reversed and the tank system layout only exists after the visual designer has selected the graphic components and placed them in this order. These graphical components do not have predecessor/successor descriptions stored in them to require an order based on a material flow and prevent mistakes, is respectfully traverses because Elmqvist architecture for automation is based on sequence control (abstract lines 2-4) i.e. a sequence of events or processes. The architecture graphically connects the predefined modules of the system (System overview, lines 8-10). Also the automation code or program is generated based on the physical objects that are present in the plant (connecting of the object and sequence of processing includes the material flow, see Object and data flow based language), the physical objects doesn't

come into existence after the designer selects them, but the designer's code is based on the physical objects as in present application's layout of plant.

V. predecessor/successor relationship in view of Juras: Appellant's argument that Juras is not related to code generation in manufacturing process is traversed because in accordance with the Patent examining policies and procedures, References from different fields of endeavor may be combined under an obviousness rejection based on design incentives, market forces etc. predictable to a person of ordinary skill in the art. Also as discussed above primary reference Burgess describes the directed predecessor successor relation between components of system reasonably and here Juras teaches the code is generated for plant design based in the already existing information as layout of the plant.

D Leisten teaching predecessor/successor relationship

Appellant argues that Leistan has nothing to do with manufacturing and processing plant and it does not fill the deficiencies of the other references. Examiner respectfully point out that a reference that is not implemented in same process as manufacturing but teaching a key feature of defined predecessor/successor relationship as in column 3, lines 1-12; and column 20, lines 40-46, is relied upon to dynamically reflect the process or project changes and to define the way the process in done in the system for generation of code.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Hirdepal Singh/

Examiner, Art Unit 2611

Conferees:

/Shuwang Liu/

Supervisory Patent Examiner, Art Unit 2611

/CHIEH M FAN/
Supervisory Patent Examiner, Art Unit 2611